

Application Serial No. 09/431,449  
Amendment dated: August 7, 2006  
Reply to Office Action dated: January 26, 2006

### **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently amended) A method for high speed interprocess communications comprising the steps of:

detecting a previously created shared region of RAM;

if a shared region of RAM is not detected, creating and configuring a shared region of RAM for storing accumulated data;

attaching first and second processes to a message buffer in the ~~the~~ [[a]] shared region of random access memory (RAM) exclusive of operating system kernel space, each said process having a message list that is a message queue;

accumulating message data from said first process in a location in said message buffer;

said first process adding to said message list of said second process a memory offset corresponding to said location in said message buffer; and,

manipulating in said second process said accumulated data at said location corresponding to said offset,

whereby said accumulated message data is transferred from said first process to said second process with minimal data transfer overhead.

2. (canceled)

3. (canceled)

4. (Currently Amended) The method according to claim 1 ~~[[3]]~~, wherein the adding step comprises the steps of:

retrieving a memory offset in said message buffer corresponding to said location of data accumulated by said first process; and,

Inserting said memory offset in said message queue corresponding to said second process.

Application Serial No. 09/431,449  
Amendment dated: August 7, 2006  
Reply to Office Action dated: January 26, 2006

5. (Original) The method according to claim 4, wherein the inserting step comprises the step of atomically assigning said memory offset to an integer location in said message queue corresponding to said second process.

6. (Previously presented) The method according to claim 1, wherein said manipulating step comprises the steps of:

identifying a memory offset in said message list corresponding to said second process;

processing in said second process message data stored at a location in said message buffer corresponding to said memory offset; and,

releasing said message buffer.

7. (Currently amended) A method for configuring high speed interprocess communications between first and second processes comprising the steps of:

creating and configuring a message buffer in a shared region of RAM exclusive of operating system kernel space and disposing a message buffer in a shared region of random access memory (RAM) shared between said first and second processes;

accumulating message data from said first process in a location in said message buffer;

creating a message list in said shared region of RAM, whereby said message list is a message queue and can store memory offsets of message data stored in said message buffer;

said first process adding to said a message list corresponding to said second process a memory offset corresponding to said location in said message buffer; and,

manipulating in said second process said accumulated message data stored in said message buffer at a location corresponding to said offset,

whereby said accumulated message data is transferred from said first process to said second process with minimal data transfer overhead.

8. (Canceled)

9. (Canceled)

Application Serial No. 09/431,449  
Amendment dated: August 7, 2006  
Reply to Office Action dated: January 26, 2006

10. (Currently Amended) The method according to claim 7 [[9]], wherein the adding step comprises the steps of:

retrieving a memory offset in said message buffer, said memory offset corresponding to said location of said message data accumulated by said first process; and,

inserting said memory offset in said message queue corresponding to said second process.

11. (Original) The method according to claim 10, wherein the inserting step comprises the step of atomically assigning said memory offset to an integer location in said message queue corresponding to said second process.

12. (Previously presented) The method according to claim 7, wherein said manipulating step comprises the steps of:

identifying a memory offset in said message list corresponding to said second process;

processing in said second process said accumulated message data at a location in said message buffer corresponding to said memory offset; and,  
releasing said message buffer.

13. (Currently amended) A computer apparatus programmed with a set of instructions stored in a fixed medium for high speed interprocess communications, said programmed computer apparatus comprising:

means for detecting a previously created shared region of RAM;

means for creating and configuring a shared region in RAM for storing accumulated data if a previously created shared region of RAM is not detected by said detecting means;

means for attaching first and second processes to a message buffer in the a shared region of random access memory (RAM) exclusive of operating system kernel space, each said process having a message list that is a message queue;

means for accumulating message data from said first process in a location in said message buffer;

Application Serial No. 09/431,449  
Amendment dated: August 7, 2006  
Reply to Office Action dated: January 26, 2006

means for said first process to add to said message list of said second process a memory offset corresponding to said location in said message buffer; and,

means for manipulating in said second process said accumulated data at said location corresponding to said offset.

14. (Canceled)

15. (Canceled)

16. (Currently Amended) The computer apparatus according to claim 13 [[15]], wherein the adding means comprises:

means for retrieving a memory offset in said message buffer corresponding to said location of data accumulated by said first process; and,

means for inserting said memory offset in said message queue corresponding to said second process.

17. (Original) The computer apparatus according to claim 16, wherein the inserting means comprises means for atomically assigning said memory offset to an integer location in said message queue corresponding to said second process.

18. (Previously presented) The computer apparatus according to claim 13, wherein said manipulating means comprises:

means for identifying a memory offset in said message list corresponding to said second process;

means for using in said second process message data at a location in said message buffer corresponding to said memory offset; and,

means for releasing said message buffer.

19. (Previously presented) The method according to claim 1, further comprising the step of locking said accumulated data to prevent said first process from accessing said accumulated data while said accumulated data is being manipulated.

20. (Previously presented) The method according to claim 13, wherein said accumulated data is locked to prevent said first process from accessing said accumulated data while said accumulated data is being manipulated.

{00011746;}